

REMARKS

I. Introduction

These remarks are set forth in response to the Non-Final Office Action. As this amendment has been timely filed within the three-month statutory period, neither an extension of time nor a fee is required. Presently, claims 1 through 29 are pending in the Patent Application. Claims 1, 11 and 20 are independent in nature. In the Office Action, each of claims 1 through 29 has been rejected under 35 U.S.C. § 102(b) as being anticipated by United States Patent No. 6,173,266 to Marx et al. (Marx). In response, the Applicants have amended claims 1, 11 and 20 to expressly define the term "catch event".

II. The Applicants' Invention

The Applicants have invented a method, system and apparatus for creating and defining standard catch styles in order to simplify a programmer's task in managing standard catch events while generating speech application code such as, for example, VoiceXML source code. As defined in Paragraph [0004] of the Applicants' specification, a catch event is an event such as a user request for help, a non-input entry, and a non-matching entry in which the user entry is not understood, that may occur during a given dialog turn. In the invention, an interface may be presented to a programmer or application developer that allows him or her to select one of a number of different catch "styles" where each "style" provides a different level of complexity with regard to preparing the system's audio response played in a typical dialog turn. A dialog turn, in this case, is initiated upon the occurrence of a standard catch event, where a standard catch event in an interactive voice application is defined as user requests for help, or a no-input or no-match event.

III. Rejections on the Art

A. Characterization of the Marx Reference

In Marx, dialogue modules are provided, each including executable code enabled to accomplish a predefined interactive dialogue task in an interactive speech application. A graphical user interface then can be provided to represent the dialogue modules as icons in a graphical display in which icons for a subset of dialogue modules are selected in the graphical display in response to user input. The icons for the subset of dialogue modules can be graphically interconnected into a graphical representation of a call flow for the interactive speech application, and the interactive speech application can be generated based upon the graphical representation. Using the graphical display, configuration parameters can be associated with specific dialogue modules, each configuration parameter causing a change in operation of the dialogue module when the interactive speech program executes. As such, a window can be displayed for setting the value of the configuration parameter in response to user input, when an icon for a dialogue module having an associated configuration parameter is selected.

B. Response to Examiner's Arguments

Claim 1 of the Applicants' patent application as amended recites:

1. (Currently Amended) A method of defining standard catch styles used in generating speech application code for managing catch events, the method comprising the steps of:

presenting a style-selection menu that allows for selection of one or more catch styles, each catch style corresponding to a system response to a catch event, *the catch event comprising at least one event in which a user entry is not understood occurring during a dialog turn*, the event being selected from the group consisting of a user request for help, a non-input entry, and a non-matching entry; and,

upon selection of a catch style, preparing the system response for each catch event.

The Applicants direct the Examiner's attention to the expressly recited portion of the amended claim, "the catch event comprising at least one event in which a user entry is not understood occurring during a dialog turn".

In Marx, no such response to a "catch event" is provided. Rather, in Marx, specifically, Figures 9 and 11 and the accompanying text of column 18, lines 17 through 29, only a dialog of features which can be enabled for a module instance is described. Those features include barge in threshold and beep after prompt, to name a couple. The complete text of column 18, lines 17 through 29 is provided herein for the convenience of the Examiner.

Selecting Features 930 from the window of FIG. 9 opens a window such as that illustrated in FIG. 11, displaying information about the various features that may be enabled in a specified Dialogue Module instance 850. The features shown in FIG. 11 include an initial prompt, whether to enable "barge-in" handling, setting a barge-in threshold (how loud the caller must speak to enable barge-in handling), and whether to enable Beep After Prompt (playing a beep after the prompt to signal caller to speak). Parameters for these features are initially set based on configuration information provided in the Baseline 820 and System 830 Libraries, but may be overridden by parameters entered by a developer in boxes 1110-1040.

Thus, no discussion of a "catch event" can be found in connection with Figures 9 and 11.

Notably, portions of Marx do address the notion of "error recovery" and audio prompts used to address encountered errors in a call flow. As described in column 20, line 17 to column 21, line 8 of Marx, however, the mechanism for programming error recovery prompts is to custom code the "prompt file" for a service or to embed the prompt directly in the dialog module instance. A verbatim reproduction of column 20, line 17 through column 21, line 8 of Marx is provided again for the convenience of the Examiner:

d. Error Recovery Parameters

Selecting the Error Recovery option 950 from the window 900 of FIG. 9 opens a window 1600 such as that illustrated in FIG. 16, which allows a developer to customize the error recovery parameters to determine the call flow within a Dialogue Module instance. As noted above with reference to FIG. 6 and as shown in FIG. 16, error recovery parameters that can be customized include the timeout period, the maximum number of times a Dialogue Module will allow consecutive timeouts, the maximum number of times the Dialogue Module will allow consecutive

recognition errors in understanding a caller's responses to a specific prompt, confirmation options, and fallback options. Default values for these parameters are initially provided by configuration information in the Baseline 820 and System 830 Libraries, and may be customized for specific instances of Dialogue Modules using the GUI such as the window 1600 of FIG. 16.

Other error recovery parameters include the content of apology prompts and reprompts. FIG. 8 shows a set of prompt files 822, 832 stored in the Baseline 820 and System 830 Libraries. *Such files include files in the appropriate format for standard prompts such as timeout apology prompts, error apology prompts, reprompts, and success messages. Customized prompts can also be provided in the prompt files, or stored elsewhere where they may be accessed within Dialogue Module instances.*

As noted above, a variety of prompt may be provided in addition to the initial prompt, including, for example, a first and second timeout apology prompt, a first and second error apology prompt, as well as general reprompt prompts. The configuration data for Dialogue Module Templates 810 provided by the Baseline 820 and System 830 Libraries may include default prompts including a first timeout apology prompt of "I'm sorry, I didn't hear your response" and a second timeout apology prompt of "I'm sorry, I still couldn't hear your response," a first error apology prompt of "I'm sorry, I didn't understand what you said" and a second error apology prompt of "I'm sorry, I'm still having difficulty understanding you." The default prompts may also include a first general reprompt of "Please say your answer C now," a second general reprompt of "Please say your answer again," and a default success prompt of "Thank you."

As noted above, the prompts may be specified in any appropriate format. For example, some embodiments may allow the prompt to be specified by its filepath, by a given name (for example, if named and stored in the Prompt Files 822, 832 shown in FIG. 8), or if a text-to-speech synthesizer is used, by its text.

Some templates, such as the ItemList Module template, require the developer to create at least some of the prompts, using an appropriate Service to create and save the prompts so that they can be properly output to callers. For example to customize an existing prompt, a developer can open the prompt file in an appropriate Service and modify the prompt. To provide a new prompt, the developer can create a new prompt file and identify its filepath to Dialogue Module instances that output that prompt. Alternatively, in systems using text-to-speech synthesizers, a developer can simply provide the text of the prompt to a Dialogue Module instance.

The emphasized portion of the excerpt above demonstrates that a number of prompts are provided for response to various error conditions. However, no where in the excerpt is it suggested that a "*style-selection menu*" is presented "that allows for selection of one or more catch styles, *each catch style corresponding to a system response to a catch event*". So much is required by the express language of claims 1, 11 and 20. Moreover, the Applicant has amended the independent claims to particularly define "catch event" consistently with paragraph [0004] of Applicants' specification leaving no question as to the meaning of the term catch event. Thus,

the Applicants respectfully observe that Marx cannot support a prima facie case of anticipation under M.P.E.P. 2131 in which it is stated, "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

IV. Conclusion

Thus, the Applicants believe that claims 1-29 distinguish over the cited art and stand patentable and ready for an indication of allowance. To that end, the Applicants respectfully request the withdrawal of the rejections under 35 U.S.C. § 102(b) owing to the foregoing remarks. This entire application is now believed to be in condition for allowance. Consequently, such action is respectfully requested. The Applicants request that the Examiner call the undersigned if clarification is needed on any matter within this Amendment, or if the Examiner believes a telephone interview would expedite the prosecution of the subject application to completion.

Respectfully submitted,

Date: August 3, 2007

/Steven M. Greenberg/
Steven M. Greenberg
Reg. No.: 44,725
Attorney for Applicant(s)
Carey, Rodriguez, Greenberg & Paul, LLP
950 Peninsula Corporate Circle, Suite 3020
Boca Raton, Florida 33487
Customer No. 46322
Tel: (561) 922-3845
Fax: (561) 244-1062